

Introduction to TouchDesigner

NVOID ART-TECH LIMITED

目录

序言	3
1 基础知识	7
1.1 信号流和连接线	7
1.2 创建 OP 元件.....	8
1.3 鼠标和键盘导航	11
1.5 使用外部文本编辑器.....	13
1.6 帮助	14
2 用户界面介绍	15
2.1 参数窗口.....	15
2.2 参数.....	17
2.3 运动控制.....	22
2.4 时间轴设置	23
2.5 分屏显示	24
2.6 资源查看器	27
2.7 搜索对话框	29
2.8 实时渲染	30
2.9 有用的快捷方式.....	31
3 TOP 元件	32
3.1 介绍	32
3.2 Movie 元件	33
3.4 Null TOP 和 Select TOP	35
3.5 编码器.....	36

序言

依靠无数的技术和技巧，一个工具软件，才在艺术家手中得以创造出优秀的作品。自从我听说 Elburz 打算写本关于 TouchDesigner 的书，就一直对本书充满兴趣。

Elburz 对 TouchDesigner 有着深厚的知识储备和深刻的理解，这是源于多方面的。比如论坛上对无数用户的帮助和交流；一些时间紧迫的重大项目的实施；用测试版的 TD 与我们 Derivative 公司一起工作；以及他纯粹的探索精神。

本书会是一本很好的办公桌或床边读物，一些之前我一直没有机会提出的一些 TD 背后的技术和概念，Elburz 在这里都提了出来。所以，本书既适合从头到尾细读，也可以跳着随便翻翻，像我就常能在里面发现宝贵的知识和深刻的见解。

非常感谢 Elburz 的这一工作丰富了 TD 社群。我相信它会产生巨大的反响。

Greg Hermanovic

Derivative 公司创始人

作者的话

写这本书有两个目的：

教一教 TD 088 的基础知识；给新手们创建一个群体协作型资源

第一个目的很直接。我们将看到各种 UI 元素，聊聊 OP 元件，探索逻辑工作流程，网络优化，接口性能，显示管理等等。我们将解释论证 TD 的核心元素，许多经常会遇到的问题将在这被提前处理。

经过本书的学习，我们将学习如何通过实例工程和视频教程，着手解决一些问题。这些课程将会很实用，我们将会从基础做起，汇总一些工程和有用的项目和模块。

第二个目的是对社群的要求稍高一点。我们真的相信这个社群，所以我们不是拿它来卖，而是希望每个人都能够免费使用本书（包括文本、视频和示例文件）。沿着这个想法，我们想更进一步，不仅让每个人能免费获取本书内容，而且每个人都可以自由的为文书添砖加瓦。

这意味着，任何人都可以随意添加、注释、更改、标记、删除或增加本书中的资源。全书已经用 LaTeX（译者：排版软件）写完，书中的代码，项目文件，例子，图片，视频，以及它所包含的一切，都将被托管在 GitHub 上（译者：托管代码网站），并将免费-在 Creative Commons（译者：一个知识共享组织）许可下-供任何人下载，分享和修改。

处于对资源质量的考虑，虽然每个人都能够对本书的知识库重新编辑，但我们会在整合进主资源库前审核所有的变更。所有贡献的人将被致谢，并加入到本书的“编辑人”部分。这样，一个新的用户只需要找一个 PDF，或下载链接，就能收到 TD 圈的最新知识。

我们真心希望我们 TD 社群能从事好这项工作，创建完善的资源，让初学者能从头学起。

写本书之前，清楚这本书的定位是很重要的。这本书不是 OP 参考手册。我们不会涵盖每一个参数或者用上每个 OP 元件。这本书不是用来替代 Derivative 维基上的资料，当作主要的参考资料来用的。我们将只使用并学习我们必须的东西。这本书并不能替代论坛上大量的有帮助的用户和元件。

最后，这些资源，是送给 TD 程序员和 Derivative 公司员工的礼物。是他们，无论在论坛上，还是在现实生活中，帮助我们走到今天。我们希望这个传统能与 TD 一起长存。

Elburz Sorkhabi & nVoid 艺术科技有限公司

TouchDesigner 是个啥？

许多使用者在开始时都曾花时间来回答这个问题。你会惊讶于用它来创建和执行简单的任务所花费的时间，也会惊讶于需要花费不少时间来实现一些在其他软件中已经现成的功能。

那么到底什么是 touchdesigner？答案很简单：touchdesigner 是一个视觉化的基于节点的编程语言。

让我们先从最重要的开始介绍，TD 是一种编程语言。TD 不是一个 APP 应用程序，像一般程序那样可以很容易的开始执行动作。TD 是一个很复杂深奥的环境，额，并且有不少坑。通过一段时间的实践，很多功能都能较快实现。因为本书的目标是创建可重复使用用的模块，所以一个项目从一张白纸到完工的速度将会越来越快。然而 TD 仍然只是一个编程语言。仍需我们付出时间和努力来完成一个个任务。相当一部分可能还很有些让人头疼，所有这些都需要具备基本的解决问题的能力。

我要介绍的第二点是，touchdesigner 是基于节点的图形化编程软件。这意味着，我们不用打开一个文本文档然后一行行敲代码，TD 用图形化界面来通过节点创建程序。TD 的每一个节点或 OP 元件，执行一个具体的，小的，独立动作。实现一个整体功能，需要多个节点共同完成。它们的输入和输出连接在一起，来传递信息。有许多基于节点的编程语言，如 Cycling 74 的 Max/MSP，而让 TD 与众不同的是它的视觉特性。

TD 中的一切都有可视化的对应。所有 OP 元件都有查看器。一切的一切，无论是文字、音频、视频、控制数据等，以及执行的 OP 元件，都是可视化的。这点不同于任何传统的甚至是基于节点的编程语言，正是它让 TD 成为一个不可思议的世界。良好的视觉化相关流程的能力，实现每一步的方式，都将大大简化你学习如何实现整体任务。

获取本书

有多种方式来获取 TD 入门 088 版。由于已将本书转到 gitbooks，现在你可以从下面的链接下载 PDF，EPUB，或移动版的本书：

<https://www.gitbook.com/book/nvoid/introduction-to-touchdesigner/details>

然后记得从我们的网站下载软件包，其中包括一个 PDF 的书，所有的例子，.toe 文件和一些跟视频教程同步的项目文件。

链接是：

https://d31vryd1jmut49.cloudfront.net/Introduction_to_TouchDesigner.zip

高清视频教程在 Vimeo（译者:国外的优酷）的一个独立频道。所有的文件都可以从 Vimeo 下载来在线下观看。（译者: 然而你需要翻墙）

下面是链接：

<https://vimeo.com/channels/845218>

你也可以从 GitHub 下载源代码，用其中的任意部分编译。你可以通过对资源文件的重新分配来自由更改、编辑、添加、删除和修改文本。编辑完成后，提交你的更新，它们将通过我们的管理评审，并合并到主分支，你将因此被放到贡献者名单里，和你的贡献一起添加进本书。

有关此资源更多的一般信息，访问 <http://book.nvoid.com>

编辑本书

与以前的 LaTeX 版本不同，不必建立环境就能投稿本书。你可以使用普通的 Markdown（译者:标记性排版语言），你的更新会被资源库保存，gitbookS 会自动创建一个新版的网页、PDF、EPUB（译者:电子书）和手机版的书。

归属许可

本资源的授权依照国际 Creative Commons - attribution-noncommercial-sharealike-4.0 标准。

链接：<http://creativecommons.org/licenses/by-nc-sa/4.0/>

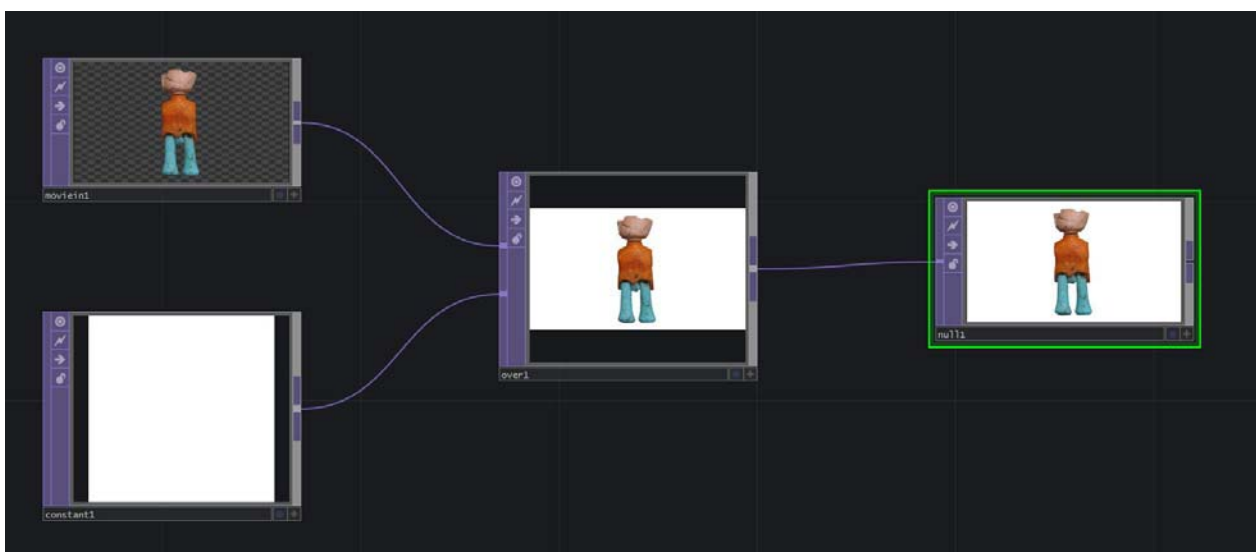
Thank you!

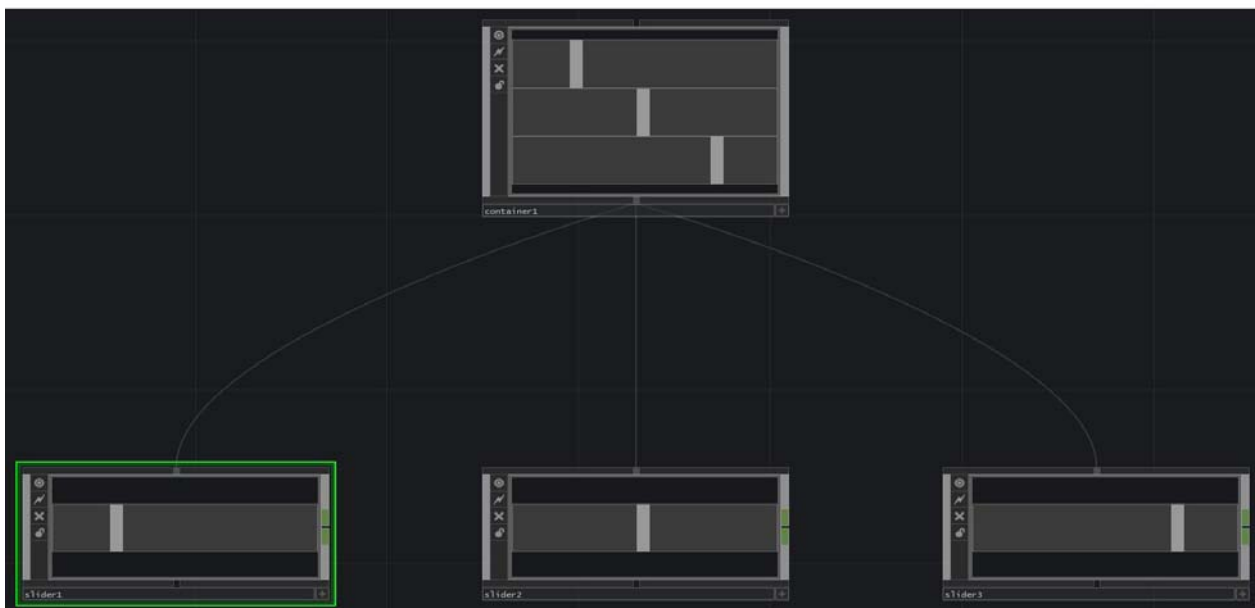
1 基础知识

1.1 信号流和连接线

连接 OP 元件是 TD 中最基本的操作。所有项目都是无数的 OP 元件连线组合而成的。每个 OP 元件自身完成一个特定的功能，当他们组合成网络，就能解决非常复杂的问题。

TD 中的数据都是从左往右传递。所有 OP 元件总是左边输入，右边输出。输入输出同时也遵循按顺序执行，从上往下。下图中，沿着左边开始的两个信号，从左到右，组件一个个被整合。有趣的是，这些从左往右拥有相同信号流的 OP 元件，依然遵从父子关系，从上往下传递。最上面是祖宗，最下面是孙子。再往下那张图的例子中，有一个由一些滑块组成的界面。在那张图中，三个滑块就是容器的子物体。





1.2 创建 OP 元件

创建 OP 元件有多种方法。每种方法都有它适合的使用场景。最简单的两种方法是点击“Tab”键，或者在背景界面上双击。

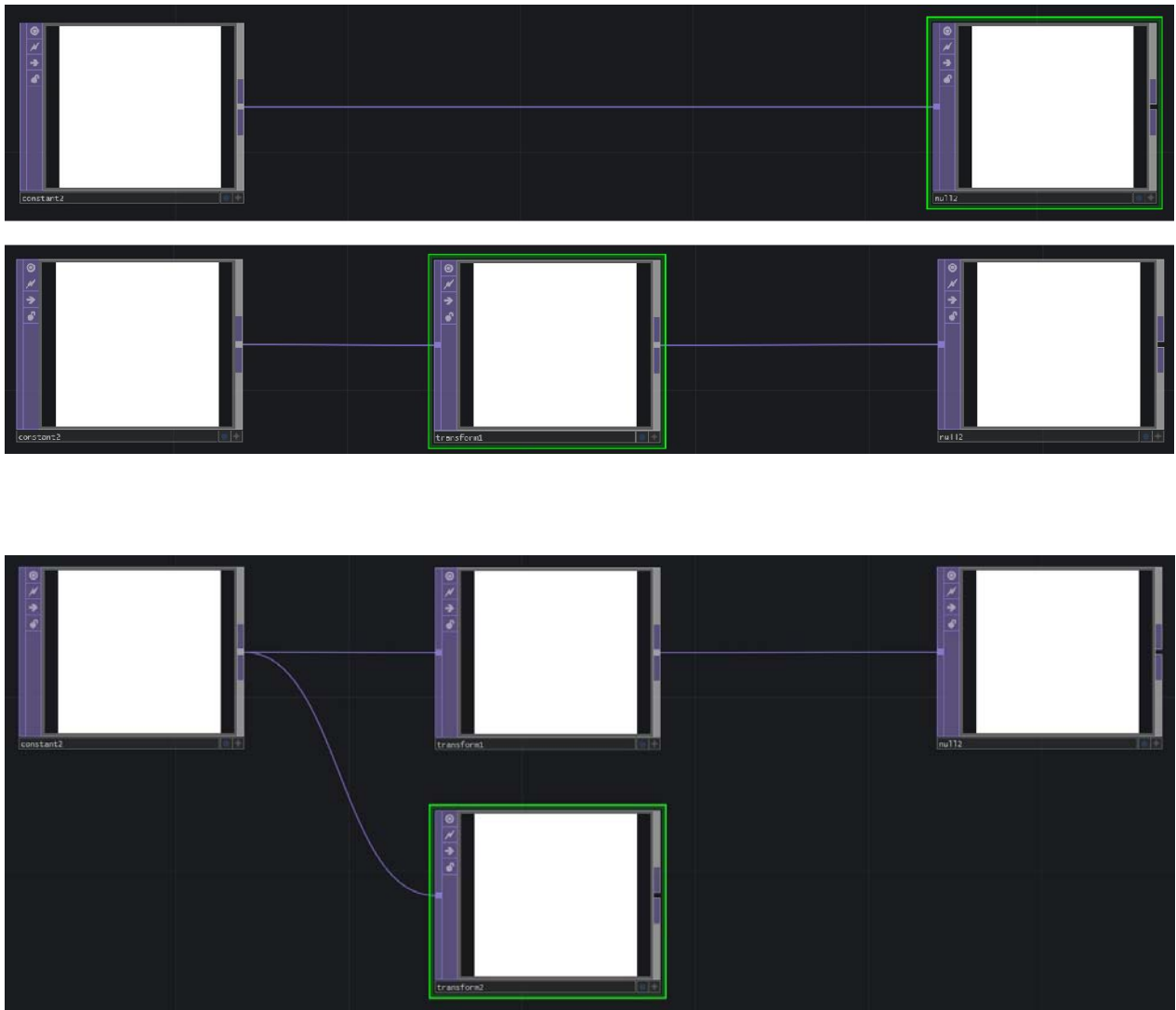
对已经有了连接线的 OP 元件，有两种方法添加相关联的 OP 元件。一种是右击现有 OP 元件的输出或者输出，这样就会直接创建关联性的新元件。当往一个已有的链路中添加元件时，这种方法尤其好使。

例如，有一个 Constant TOP 元件连接到一个 Null TOP 元件，这时，想在连个元件之间加入一个 Transform TOP。右键单击 Constant TOP 的输出端，或 Null TOP 的输入端，选择 Transform TOP，就在这俩中间插入 Transform TOP。

另一个办法是在现有 OP 元件的输出端点击鼠标中键，区别在于，右键创建的 OP 会自动插入已有的两个 OP 中间，而中键创建的 OP 会在原链路外，新建一个并联的分支。

通过右键连接线，点击“Insert Operator”或“Add Operator”，也能取得相似的效果。“OPInsert Operator”相当于右键 OP 的输出端口，并把它连入当下 OP 的链路，而“Add Operator”的作用就像是点击鼠标中键，创建一个平行于现有链路的分支。

在下图中的例子里，有一个 Constant TOP 和一个 Null TOP,再往下那张图里，这两 TOP 之间的连接线被右键单击后，通过插入 OP 的方式，创建了一个 Transform TOP。最后那张图里，是通过右键连线，用“Add Operator”（增加 OP）的方式创建的 Transform TOP。注意它是如何创建的平行于第一个 Transform TOP 的接线。



在创建 OP 时，有 2 个超有用的快捷键：“ctrl”和“shift”。打开创建 OP 对话框，按住“ctrl”，然后连续选择多个 OP，这些 OP 会按顺序从上到下添加进界面。这招可以把几个 OP 快速添加进项目。

按 SHIFT 也是一样的操作方法。区别是，这样创建的 OP 会从左往右直接串联好，这种方法可以快速创建已经连号线的若干 OP。

这两快捷键真心强大。当用来创建一个连接好的 OP 时，尤其好使。举个例子，一个项目需要 3 条 OP 链。

第一条先是一个 Null TOP，后面接一个 Blur TOP，最后面接个 Circle TOP。

第二条先后是 Noise TOP, Edge TOP, Circle TOP。

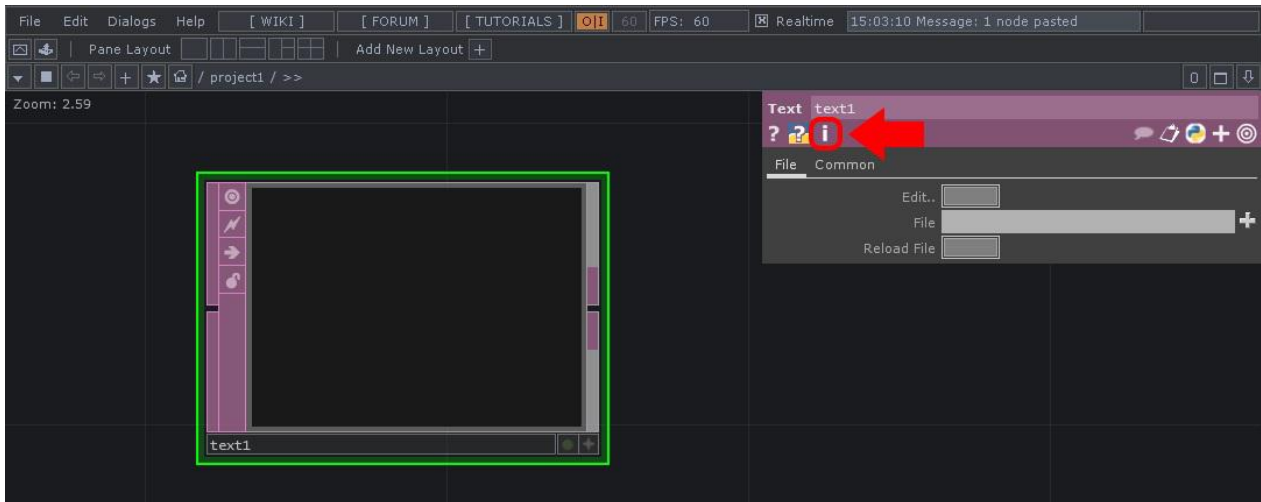
第三条是 Null TOP, Blur TOP, Movie In TOP。

咱们一步步过一遍这个步骤，来演示下这两快捷键的实际用法。

- 1.打开 OP 对话框
- 2.按住 Shift
- 3.单击 Circle TOP, Blur TOP, Null TOP , 创建第一个链路。
- 4.释放 “Shift” 键
- 5.按住 “ctrl” 。这将把下一个 OP 放在第一个 OP 下面。
- 6.保持按住 ctrl,点击 Circle TOP
- 7.释放 “ctrl” 键
- 8.按住 “shift” 键
- 9.点击 Edge TOP, Noise TOP, Null TOP,创建第二链路。
- 10.释放 Shift
- 11.按下 “ctrl”
- 12.同时 , 点击 Movie In TOP。
- 13.释放 “ctrl” 键
- 14.按下 shift
- 15.点击剩余的那俩 OP : Blur TOP, and Null TOP。
- 16.现在所有的 OP 都创建了 , 使用 “ESC” 键关闭 OP 创建对话框。

关闭 OP 创建对话框后 , 所有需要的 OP 已经被添加进项目 , 并连好接线。这些快捷键不光省得你创建每个 OP 元件都要打开关闭一次 OP 对话框 , 而且省的你给他们连线。

1.3 鼠标和键盘导航



点击图中的 i，可以获得更多所选 OP 元件的详细信息。

在 TD 工程中导航，有一些快捷键。其中两个是 i 和 u 键。

u 键会跳出当下组件，进入上一层网络。

i 键相反，进入一个网络或组件。

如果想看到某个网格中的所有 OP 元件，按 h。它相当于执行当下网格的 home 动作。

1.4 网络和路径

所有的 TD 工程都是由网络组成的。一个网络其实就是一堆 OP 元件。

网络里被封装了各种组件，比如 Container 组件，Base 组件，Geometry 组件等等。

网络这玩意能无限嵌套。

最上面那层，叫根层。

在根层里，会找到 TD 系统和 UI 元素。

你最好训练自己从项目一开始，就习惯于封装和组织网络。在网络编辑器的顶部，有个路径栏，里面会显示当下路径。



所有 OP 都有路径。这种路径和 Unix 文件的路径相似。

路径有两种，“绝对路径”和“相对路径”。

绝对路径是项目的根目录或者“/”。

相对路径是从一个 OP 指向另一个 OP 的路径。它开始于网络中引用的某个 OP,而不是根目录。

打开范例“Paths.toe”这个例子可以说明路径是咋回事。

TD 项目是从工程根目录下的那个叫“network1”的 Container 组件开始的。在 network1 里，有两个 OP。

“rel1”是个 Text DAT 元件。它的内容语句中，写里有两条路径。第一个是绝对路径。它始于根目录，指向元件。另一个是相对路径，它始于本元件，指向 network2 中的另一个 Text DAT，“rel2”。相对路径从当下位置指向目的地。从 rel1 到 rel2，路径只用穿过 network2，所以就是‘network2/rel2’。

注意 network2 显示的是它内部的某个元件，这一点将在后面的例子中详细讨论。现在我们要关心的是它的路径的用法。在 network2 的 Operator View 的 there 参数中，有一条路径‘./display’。display 是它内部某个元件的名字，‘./’则表示当前元件也就是 network2 的下一层。

在 network2 中，display 的内容被显示在它的上层容器中，另外两个元件则演示了路径的其他用法。

abs1 是绝对路径的一个例子。rel2 演示了它与 abs1 之间的相对路径，以及它与上层网络中元件 rel1 的相对路径。

rel1 是元件名称，../表示位于上一层网络。../可以叠加使用直到根网络，但还有别的更高效的方法。

1.5 使用外部文本编辑器

TD 中本身可以创建和编辑简单的 Python 脚本。不过随着脚本增长，用外部的编译器会更省事快捷。这样会有一大堆好处，我们简单列举：

- 代码行号；
- 分颜色的代码；
- 查找替换功能；
- 自动联想语句；

所以，在 TD 中广泛使用 Python,会让你的工作更高效。

在编辑菜单中打开参数对话框，进入 DATs 参数选项，在 “Text Editor” 设置中，点击文件浏览的图标，找到并选定编辑器的 exe 文件，打开，点击接受。

设置完成后，右键一个 DAT，然后点击 “Edit Contents” 选项，DAT 就会用刚才设置的编辑器打开。

有一个 “Table Editor” 参数，专门用来设置 Table 类型 DATs 的编译器。

这有两 TD 中用的比较多的跨平台编译器：

Sublime Text 3 <http://www.sublimetext.com/>

Notepad++ <http://notepad-plus-plus.org/>

1.6 帮助

关于程序和具体 OP 元件的任何问题，可以参考官方的 Wiki。每个 OP 有两个快捷方式，可以指向一个 WIKI 中的网页。

它们位于参数窗口，外形都是问号的样子。其中一个介绍本 OP 元件的用法。另一个带 Python Logo 的问号，是专门介绍跟这个 OP 相关的 Python 语句的。

